UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/728,649	12/05/2003	Perry H. Wang	42P17015	8436

8791 7590 09/07/2007
BLAKELY SOKOLOFF TAYLOR & ZAFMAN
1279 OAKMEAD PARKWAY
SUNNYVALE, CA 94085-4040

EXAMINER	
TANG, KENNETH	
ART UNIT	PAPER NUMBER
2195	
MAIL DATE	DELIVERY MODE
09/07/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	**Application No.** 10/728,649	**Applicant(s)** WANG ET AL.	
	Examiner Kenneth Tang	**Art Unit** 2195	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *17 June 2005 and 12/5/03*.

2a)□ This action is **FINAL**. 2b)☒ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-27* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-18 and 20-27* is/are rejected.

7)□ Claim(s) *19* is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)□ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *05 December 2003* is/are: a)☒ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☒ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date *6/17/05*.

4) □ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) □ Notice of Informal Patent Application

6) □ Other: _____.

DETAILED ACTION

1. Claims 1-27 are presented for examination.

Claim Objections

2. Claim 25 is objected to because of the following informalities: There is an empty space between "resumed" and the following semicolon. Therefore, on claim 25, line 2, "resumed ;" should be amended to "resumed;". Appropriate correction is required.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

3. **Claims 1, 3-4, 10-12, 14-15, 17-18, 20-21, 23-25, and 27 are rejected under 35 U.S.C. 103(a) as being unpatentable over Ozur et al. (hereinafter Ozur) (US 5,247,676) in view of Hogle et al. (hereinafter Hogle) (US 6,560,626 B1).**

4. As to claim 1, Ozur teaches an apparatus comprising:

a trigger-response mechanism (spawning a helper thread in the event a procedure call is made) that includes at least one bank of user-programmable registers (software thread possessing register set) (col. 1, lines 15-20); and

a thread switch handler coupled to the trigger-response mechanism (RPC run-time facility 84-1 and 84-2 handles the context switching between threads) (col. 6, lines 23-32 and 40-62), the thread switch handler to invoke a second instruction stream (helper thread) responsive to an indication from the trigger-response mechanism that a trigger event has occurred during execution of a first instruction stream (spawning a helper thread from a main thread in the event a procedure call is made) (col. 1, lines 31-37).

5. By definition, a thread is a stream of executable code within a process. Therefore, the main thread and helper threads of Ozur are streams of executable code within a process. Applicant's Specification also discloses its main thread and helper threads having streams of instructions (see [0025], [0033], etc.). Also by definition, all threads within a process share process instruction, code & data segment, open file descriptor, signal handler, user ID, and group ID. Threads have their own set of registers including program counter and stack pointer.

6. Ozur is silent in teaching the trigger events being user-defined. However, Hogle teaches a context/thread switching based on user-defined events (col. 1, lines 14-28). As taught in Hogle, a thread's context contains information such as its pointers, and register values, etc. The thread's context is saved before the switch and then restored to the next thread being switched to (col. 5, lines 24-39). Ozur and Hogle are analogous art because they are both in the same field of endeavor of a context switching within a multithreaded environment. One of ordinary skill in the

art would have known to modify the context switching of Ozur to include the switching based on user-defined events such as in Hogle. The suggestion/motivation for doing so would have been to provide the predicted result of being able to switch based on input from the user, thus giving the user more control (col. 1, lines 14-28). For example, user could select the timeout period for the wait function to set how long the thread waits before resuming execution (col. 1, lines 32-45). Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to combine Ozur and Hogle to obtain the invention of claim 1.

7. As to claim 3, Hogle teaches wherein the thread switch handler is further to invoke the second instruction stream responsive to an indication from the trigger-response mechanism that an asynchronous user-defined trigger event has occurred during execution of the first instruction stream (see Abstract, col. 5, lines 47-57).

8. As to claim 4, Hogle teaches wherein the thread switch handler is to save an instruction pointer address for the first instruction stream before invoking the second instruction stream (col. 5, lines 24-39).

9. As to claim 10, Hogle teaches wherein the thread switch handler is to save context information for the first instruction stream before invoking the second instruction stream (col. 5, lines 27-35).

10. As to claim 11, Hogle teaches wherein: the thread switch handler is further to save context for the first instruction stream in a memory location before invoking the second instruction stream (col. 5, lines 27-35).

11. As to claim 12, Hogle teaches wherein: the thread switch handler is further to save context for the first instruction stream in a register before invoking the second instruction stream (col. 5, lines 27-35).

12. As to claim 14, Ozur teaches a system comprising:

a memory to store an instruction (register set, col. 1, lines 15-19, or non-volatile memory that is inherent in a computer system, col. 2, lines 12-14); and

a single-threaded processor coupled to the memory, wherein the processor provides a thread context (register set, col. 1, lines 15-19; CPU and non-volatile memory that is inherent in a computer system, col. 2, lines 12-14);

wherein the processor includes a trigger-response mechanism to detect a trigger event and also includes a switch handler to invoke a helper thread responsive to occurrence of the trigger event (spawning a helper thread from a main thread in the event a procedure call is made) (col. 1, lines 31-37).

13. By definition, all threads within a process share process instruction, code & data segment, open file descriptor, signal handler, user ID, and group ID. Threads have their own set of registers including program counter and stack pointer.

14. Ozur is silent in teaching the trigger events being user-defined. However, Hogle teaches a context/thread switching based on user-defined events (col. 1, lines 14-28). As taught in Hogle, a thread's context contains information such as its pointers, and register values, etc. The thread's context is saved before the switch and then restored to the next thread being switched to (col. 5, lines 24-39). Ozur and Hogle are analogous art because they are both in the same field of endeavor of a context switching within a multithreaded environment. One of ordinary skill in the art would have known to modify the context switching of Ozur to include the switching based on user-defined events such as in Hogle. The suggestion/motivation for doing so would have been to provide the predicted result of being able to switch based on input from the user, thus giving the user more control (col. 1, lines 14-28). For example, user could select the timeout period for the wait function to set how long the thread waits before resuming execution (col. 1, lines 32-45). Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to combine Ozur and Hogle to obtain the invention of claim 1.

15. As to claim 15, Hogle teaches using a system memory that is a Random Access Memory 25 (col. 3, lines 63-67 through col. 4, line 1). Ozur and Hogle lack or does not expressly disclose the system memory being a Dynamic Random Access Memory (DRAM). However, Examiner takes Official Notice that DRAM is a type of memory conventionally used in most personal

computers and is well-known. Dynamic random access memory (DRAM) is a type of random access memory that stores each bit of data in a separate capacitor within an integrated circuit. Since real capacitors leak charge, the information eventually fades unless the capacitor charge is refreshed periodically. Because of this refresh requirement, it is a *dynamic* memory as opposed to SRAM and other *static* memory. The advantage of DRAM over static RAM is its structural simplicity: only one transistor and a capacitor are required per bit, compared to six transistors in SRAM. This allows DRAM to reach very high density. Since DRAM loses its data when the power supply is removed, it is in the class of *volatile* memory devices. It would have been obvious at the time the invention was made to one of ordinary skill in the art to employ memory such as DRAM since Examiner takes Official Notice that DRAM is a type of memory conventionally used in most personal computers and is well-known in the art.

16. As to claim 17, Hogle teaches wherein: the instruction is a marking instruction that specifies the trigger event, the trigger event being asynchronous; and the trigger-response mechanism is further to detect the asynchronous trigger event (see Abstract, col. 5, lines 47-57).

17. As to claim 18, Ozur teaches wherein: the switch handler is further to maintain minimal context information for a current thread before invoking the helper thread, wherein the minimal context information excludes traditional context information (col. 4, lines 25-26 and 54-57). Applicant's Specification describes the minimal context information to being an instruction pointer address, for example ([0043]). Since Ozur teaches a call block 36 which stores

information that identifies the location of the calling thread 28 and/or storing a call block pointer 56 in a thread execution block, Applicant's definition of the minimal context information is satisfied.

18. As to claim 20, Ozur teaches wherein the minimal thread context information comprises an instruction pointer address value (col. 4, lines 25-26 and 54-57).

19. As to claim 21, Ozur teaches a method comprising:

detecting a trigger condition (spawning a helper thread in the event a procedure call is made) (col. 1, lines 31-37);

suspending execution of a first thread on a single-threaded processor (col. 6, lines 23-32);

utilizing hardware to save minimal context information for the current thread without operating system intervention (col. 4, lines 25-26 and 54-57); and

invoking a second thread on the single-threaded processor without operating system intervention (spawning a helper thread from a main thread) (col. 1, lines 31-37).

20. Ozur is silent in teaching the trigger events being user-defined. However, Hogle teaches a context/thread switching based on user-defined events (col. 1, lines 14-28). As taught in Hogle, a thread's context contains information such as its pointers, and register values, etc. The thread's context is saved before the switch and then restored to the next thread being switched to

(col. 5, lines 24-39). Ozur and Hogle are analogous art because they are both in the same field of endeavor of a context switching within a multithreaded environment. One of ordinary skill in the art would have known to modify the context switching of Ozur to include the switching based on user-defined events such as in Hogle. The suggestion/motivation for doing so would have been to provide the predicted result of being able to switch based on input from the user, thus giving the user more control (col. 1, lines 14-28). For example, user could select the timeout period for the wait function to set how long the thread waits before resuming execution (col. 1, lines 32-45). Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to combine Ozur and Hogle to obtain the invention of claim 1.

21. As to claim 23, Hogle teaches wherein: detecting a user-specified trigger condition further comprises determining that an asynchronous condition specified in a marking instruction has been encountered (see Abstract, col. 5, lines 47-57).

22. As to claim 24, Ozur teaches wherein: utilizing hardware to save minimal context information further comprises saving an instruction pointer address value (col. 4, lines 25-26 and 54-57).

23. As to claim 25, Ozur (col. 4, lines 25-26 and 54-57, col. 6, lines 23-32) and Hogle (col. 5, lines 24-39) teach further comprising: determining that the first thread should be resumed;

restoring the minimal context information for the first thread; and resuming execution of the first thread without operating system intervention.

24. As to claim 27, Hogle teaches wherein detecting a user-specified trigger condition further comprises: generating an asynchronous response to indicate that the second thread should be invoked (see Abstract, col. 5, lines 47-57).

25. Claims 2, 16, and 22 are rejected under 35 U.S.C. 103(a) as being unpatentable over Ozur et al. (hereinafter Ozur) (US 5,247,676) in view of Hogle et al. (hereinafter Hogle) (US 6,560,626 B1), and further in view of Phillips et al. (hereinafter Phillips) (US 2002/0026502 A1).

26. As to claim 2, Ozur and Hogle teach wherein the thread switch handler is further to invoke the second instruction stream responsive to an indication from the trigger-response mechanism that a user-defined trigger event has occurred during execution of the first instruction stream (see rejection of claim 14). However, Ozur and Hogle are silent in synchronous triggering of events. Applicant's Specification defines synchronous triggering of events to be the triggering of events due to detecting a particular opcode to be encountered (see Specification, [0025]-[0026]). Phillips teaches context switching, wherein an opcode instruction is detected during an execution phase of an execution pipeline ([0162]-[0163]). Phillips and Ozur in view of

Hogle are analogous art because they are in the same field of endeavor of context switching. One of ordinary skill in the art would have known to modify Ozur in view of Hogle's context switching system to include the detection of opcode of the trigger instruction (event triggering based on opcode). The motivation/suggestion for doing so would have been to provide the predicted result of a control means for switching and carrying out requests ([0027]).

27. As to claim 16, Ozur and Hogle is silent in teacheing wherein: the instruction is a trigger instruction; and the trigger-response mechanism is further to detect the opcode of the trigger instruction when the trigger instruction reaches an execution phase of an execution pipeline. However, Phillips teaches context switching, wherein an opcode instruction is detected during an execution phase of an execution pipeline ([0162]-[0163]). Phillips and Ozur in view of Hogle are analogous art because they are in the same field of endeavor of context switching. One of ordinary skill in the art would have known to modify Ozur in view of Hogle's context switching system to include the detection of opcode of the trigger instruction when the trigger instruction reaches an execution phase of an execution pipeline. The motivation/suggestion for doing so would have been to provide the predicted result of a control means for switching and carrying out requests ([0027]).

28. As to claim 22, Ozur and Hogle are silent wherein: detecting a user-specified trigger condition further comprises determining that a trigger instruction has been encountered. However, Phillips teaches context switching, wherein an opcode instruction is detected during an

execution phase of an execution pipeline ([0162]-[0163]). Phillips and Ozur in view of Hogle are analogous art because they are in the same field of endeavor of context switching. One of ordinary skill in the art would have known to modify Ozur in view of Hogle's context switching system to include the detection of opcode of the trigger instruction when the trigger instruction reaches an execution phase of an execution pipeline. The motivation/suggestion for doing so would have been to provide the predicted result of a control means for switching and carrying out requests ([0027]).

29. **Claims 5-7 are rejected under 35 U.S.C. 103(a) as being unpatentable over Ozur et al. (hereinafter Ozur) (US 5,247,676) in view of Hogle et al. (hereinafter Hogle) (US 6,560,626 B1), and further in view of Nojiri (US 5,179,685).**

30. As to claim 5, Ozur and Hogle are silent in further comprising: a task queue to receive the instruction pointer address. However, Nojiri teaches using a task queue so that task control blocks for respective blocks are linked together by register bank pointers (col. 4, lines 35-43). Nojori teaches that this results in task/context switching that can be realized without erroneous operation (col. 4, lines 43-45). One of ordinary skill in the art would have known to modify Ozur in view of Hogle's context switching system to include the feature of a task queue to receive the instruction pointer address. The suggestion/motivation for doing so would have been to allow for context switching to be realized and for the registers to be obtained without

erroneous operation, as stated in Nojori (col. 4, lines 43-45, col. 5, lines 1-22). Therefore, it would have been obvious to combine Nojori with the references of Ozur and Hogle.

31. As to claim 6, Nojori teaches wherein: the task queue further comprises a memory location (col. 4, lines 43-45, col. 5, lines 1-22).

32. As to claim 7, Nojori teaches wherein: the task queue further comprises a register (register bank or register sets) (col. 5, lines 1-22).

33. **Claims 8-9 are rejected under 35 U.S.C. 103(a) as being unpatentable over Ozur et al. (hereinafter Ozur) (US 5,247,676) in view of Hogle et al. (hereinafter Hogle) (US 6,560,626 B1), and further in view of Hugly (US 2002/0138706 A1).**

34. As to claim 8, Ozur and Hogle are silent in further comprising: a plurality of event counters coupled to the trigger-response mechanism, wherein each event counter is to detect an atomic processor event. However, Hugly teaches exception handling and context switching wherein a plurality of event counters are coupled to the switching mechanism, wherein each event counter detects an atomic processor event ([0041]-[0043], [0045]). Hugly and Ozur in view of Hogle are analogous art because they are in the same field of endeavor of context

switching. One of ordinary skill in the art would have known to modify Ozur in view of Hogle's context switching to include the Hogle's context switching using a plurality of event counters. The motivation/suggestion for doing so would have been to improve the handling of any access conflicts, thus improving control and reducing overheard, as stated in Hugly ([0008]). Therefore, it would have been obvious to combine Hugly with Ozur and Hogle to obtain the invention of claim 8.

35. As to claim 9, Ozur (col. 6, lines 23-32 and 40-62, col. 1, lines 31-37) in view of Hogle (col. 5, lines 24-39) teaches wherein the thread switch handler is further to invoke the second instruction stream responsive to an indication from the trigger-response mechanism that an asynchronous user-defined trigger event has occurred during execution of the first instruction stream (also see rejection of claim 1). Hugly teaches trigger events being based on one or more of the atomic processor events ([0041]-[0043], [0045]).

36. Claim 13 is rejected under 35 U.S.C. 103(a) as being unpatentable over Ozur et al. (hereinafter Ozur) (US 5,247,676) in view of Hogle et al. (hereinafter Hogle) (US 6,560,626 B1), and further in view of Spix et al. (hereinafter Spix) (US 6,195,676).

37. As to claim 13, Ozur and Hogle teaches further comprising: one or more user-programmable control registers coupled to the thread switch handler (see rejection of claim 1).

Ozur and Hogle are silent in teaching the value of the one or more control registers to indicate the weight of context information. However, Spix teaches context switching wherein the amount of context information in the registers are indicated and classified as lightweight, for example (col. 3,lines 9-35, col. 14, lines 66-67 through col. 15, lines 1-6). Spix and Ozur in view of Hogle are in the same field of endeavor of context switching. One of ordinary skill in the art would have known to modify Ozur in view of Hogle's context switching system to include Spix's feature of identifying weights of context information for the use of context switching. The motivation/suggestion for doing so would have been to minimize total context switch overhead and minimizing the delays and bottlenecks by classifying the amount of context information (weights) and context switching according to those weights (col. 3,lines 9-35, col. 14, lines 66-67 through col. 15, lines 1-6). Therefore, it would have been obvious to combine Spix with Ozur and Hogle to obtain the invention of claim 13.

38. Claim 26 is rejected under 35 U.S.C. 103(a) as being unpatentable over Ozur et al. (hereinafter Ozur) (US 5,247,676) in view of Hogle et al. (hereinafter Hogle) (US 6,560,626 B1), and further in view of Farley et al. (hereinafter Farley) (US 7,228,348 B1).

39. As to claim 26, Hogle teaches wherein detecting a user-specified trigger condition further comprises: receiving a marker instruction (asynchronous according to Applicant's Abstract) that specifies the trigger condition (see Abstract, col. 5, lines 47-57). Ozur and Hogle are silent in monitoring a plurality of atomic event indicators to detect the trigger condition. However, Farley

teaches monitoring trigger conditions and triggering events, wherein a trigger can relate to an atomic transaction, as an example (col. 3, lines 53-67). Farley and Ozur in view of Hogle are analogous art because they are in the same field of endeavor of event triggering. One of ordinary skill in the art would have known to modify Hogle and Ozur's event triggering so that it would include the monitoring of event triggers to detect trigger conditions of Farley. The motivation/suggestion for doing so would have been to trigger events with more abstract characteristics of a data stream (col. 3, lines 53-67). Complex and statistical characteristics of a trigger could be detected. Therefore, it would have been obvious to combine, Hogle, Ozur, and Farley to obtain the invention of claim 26.

Allowable Subject Matter

40. Claim 19 is objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Kenneth Tang whose telephone number is (571) 272-3772. The examiner can normally be reached on 8:30AM - 6:00PM, Every other Friday off.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng-Ai An can be reached on (571) 272-3756. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Kenneth Tang
Kt
9/2/07